A System Theoretical Perspective to Gradient-Tracking Algorithms for Distributed Quadratic Optimization

Michelangelo Bin, Ivano Notarnicola, Lorenzo Marconi, Giuseppe Notarstefano

Abstract

In this paper we consider a recently developed distributed optimization algorithm based on gradient tracking. We propose a system theory framework to analyze its structural properties on a preliminary, quadratic optimization set-up. Specifically, we focus on a scenario in which agents in a static network want to cooperatively minimize the sum of quadratic cost functions. We show that the gradient tracking distributed algorithm for the investigated program can be viewed as a sparse closed-loop linear system in which the dynamic state-feedback controller includes consensus matrices and optimization (stepsize) parameters. The closed-loop system turns out to be not completely reachable and asymptotic stability can be shown restricted to a proper invariant set. Convergence to the global minimum, in turn, can be obtained only by means of a proper initialization. The proposed system interpretation of the distributed algorithm provides also additional insights on other structural properties and possible design choices that are discussed in the last part of the paper as a starting point for future developments.

I. Introduction

Many optimization algorithms are iterative procedures that can be, thus, framed as discrete-time dynamical systems. Usual approaches to prove the convergence of these schemes, even though often based on descent, Lyapunov-like arguments, do not explicitly and deeply explore this system theoretical perspective. The great potential of system theory becomes more evident when noticing that several algorithms encode a feedback structure in their update laws. In this paper we propose a system theoretical interpretation of a state-of-the-art *distributed* optimization algorithm often known as gradient tracking, see, e.g., [1]–[8].

The authors are with the Department of Electrical, Electronic and Information Engineering, University of Bologna, Bologna, Italy, name.lastname@unibo.it.

The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 638992 - OPT4SMART) and from the European Project AirBorne (ICT 780960).

In this framework agents (systems) in a network cooperate to minimize the sum of local functions that depend on a common decision variable. Agents exchange information with neighbors in a given (sparse) communication graph and cannot rely on any centralized coordinating unit. We consider a simplified set-up in which the optimization problem is quadratic and the communication occurs according to a fixed and undirected graph.

Distributed optimization has received a large interest from the control community in the last decades. Early references on this topic are [9], [10] where the (sub)gradient method has been successfully combined with consensus averaging to design a distributed method. Recently, this approach has been enhanced by introducing a tracking technique based on the dynamic average consensus, originally proposed in [11], [12]. The tracking mechanism allows agents to obtain a local estimate of the gradient of the entire sum of functions, which is then used as a descent direction in the consensus-based update of the local solution estimate, see, e.g., [1]–[8].

First approaches providing a system theoretical perspective to distributed optimization algorithms are [13], [14]. A framework based on integral quadratic constraints from robust control theory is proposed in [15] to analyze and design (centralized) iterative optimization algorithms. In [16] authors propose a loop-shaping interpretations for several existing optimization methods based on basic control elements such as PID and lag compensators. The convergence of distributed optimization algorithms by means of proper semidefinite programs is, instead, discussed in [17]. A passivity-based approach is proposed in [18] to analyze a distributed algorithm with communication delays.

The contributions of this paper are as follows. We approach the design of distributed optimization algorithms as a control problem, by showing how system theoretical tools can be used to provide new insights on the existing algorithms, and new perspectives for future extensions. We develop the discussion for a simplified quadratic, unconstrained optimization problem, that allows us to rely on powerful tools from linear regulation theory. In particular, we cast the optimization algorithm design as a linear control problem aiming to steer the state trajectories toward the optimal solution, and we provide necessary and sufficient conditions for its solvability. We show that a class of gradient tracking distributed algorithms fits in the proposed framework, which, in turn, provides new insights in terms of structural properties of the controlled system. Specifically, the resulting algorithm, seen as a sparse dynamical system, turns out to be not completely reachable and this reflects on the need of a proper initialization and of the necessity of a stabilizing action in the "closed-loop" dynamics. The proposed system theoretical perspective suggests that robustness arguments, customary in control theory, can be used to extends these features also to optimization algorithms.

The paper is organized as follows. In Section II we introduce the distributed optimization set-up and recall the gradient tracking algorithm. In Section III we describe the system theoretical framework to solve a quadratic distributed optimization problem which is used in Section IV to analyze the gradient tracking algorithm.

Notation: We deal with discrete-time dynamical systems of the form $x(t+1) = \phi(x(t))$. For the sake of readability we omit the time dependency whenever it is clear from the context and we write x^+ in place of $x(t+1)$. Given a square matrix $M \in \mathbb{R}^{d \times d}$, we denote by $\sigma(M)$ its spectrum. A square matrix is said to be Schur if all its eigenvalues lie inside the open unitary disc. Given a square matrix $F \in \mathbb{R}^{d \times d}$, a set $\mathcal{V} \subset \mathbb{R}^d$ is said to be F-invariant if for all $v \in \mathcal{V}$ it holds $Fv \in \mathcal{V}$. We denote by I_d the $d \times d$ identity matrix and by 0_d the $d \times d$ matrix of zeros. The column vector of d ones is denoted by $\mathbf{1}_d$. Moreover, we define $\mathbf{I} = \mathbf{1}_N \otimes I_d$ where $\otimes$ is the Kronecker product. We omit the dimension of these objects whenever it is clear from the context. For $x \in \mathbb{R}^{n_1}$ and $z \in \mathbb{R}^{n_2}$, we denote by $\text{col}(x, z) \in \mathbb{R}^{n_1+n_2}$ their column concatenation. For $S \subset \mathbb{R}^n$ and $x \in \mathbb{R}^n$, we let $x + S := \{z \in \mathbb{R}^n \mid z = x + s, \, s \in S\}$.

II. The Distributed Optimization Framework

In this section we introduce the distributed optimization set-up and recall the state-of-the-art gradient tracking algorithm that we aim to investigate in this paper.

A. *Distributed Optimization Set-up*

We consider the following optimization problem

$$\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{N} f_i(\theta), \tag{1}$$

where, for each $i = 1, \ldots, N$, $f_i : \mathbb{R}^d \to \mathbb{R}$ is of the form

$$f_i(\theta) = \frac{1}{2}(\theta - \Gamma_i \theta_0)^\top C_i (\theta - \Gamma_i \theta_0), \tag{2}$$

with $C_i \in \mathbb{R}^{d \times d}$ symmetric and positive-definite, $\Gamma_i \in \mathbb{R}^{d \times p}$, and where $\theta_0 \in \mathbb{R}^p$ is an offset parameter whose role will be clarified later. In particular, problem (1) admits a unique solution given by

$$\theta^\star := \Sigma \theta_0, \quad \Sigma := \left(\sum_{i=1}^{N} C_i \right)^{-1} \sum_{i=1}^{N} C_i \Gamma_i. \tag{3}$$

We focus on iterative procedures to solve (1) that are *distributed*. In particular, we assume to have a network of N agents, each one having access only to partial information about the problem and exchanging information with a subset of the other agents. Distributed optimization algorithms are *local* update laws that fulfill the network constraints and allow agents to eventually converge the optimal solution $\theta^\star$. Formally, we model the network by means of a connected and undirected graph $\mathcal{G} = (\{1, \ldots, N\}, \mathcal{E})$ where $\mathcal{E} \subseteq \{1, \ldots, N\} \times \{1, \ldots, N\}$ is the set of edges. If $(i, j) \in \mathcal{E}$, then nodes i and j can exchange information (and, in fact, $(j, i) \in \mathcal{E}$). We denote by $\mathcal{N}_i := \{j \in \{1, \ldots, N\} \mid (i, j) \in \mathcal{E}\}$ the set of neighbors of node i in $\mathcal{G}$. We assume that $\mathcal{N}_i$ contains i itself. As usually done in consensus-based approaches, we consider a matrix $A \in \mathbb{R}^{N \times N}$ matching the graph $\mathcal{G}$, i.e., (i, j)-th entry $a_{ij} > 0$ for $(i, j) \in \mathcal{E}$ while $a_{ij} = 0$ otherwise. Moreover, A is row stochastic if $A\mathbf{1}_N = \mathbf{1}_N$, while it is column stochastic if $\mathbf{1}_N^\top A = \mathbf{1}_N^\top$. It can be proved that the spectrum of a row (or column) stochastic matrix lies in the closed unitary circle and the largest (in norm) eigenvalue is 1 and is simple.

In this paper we assume that each agent i maintains a local quantity $x_i \in \mathbb{R}^d$ representing its guess of the optimal solution $\theta^\star$, and it has only access to gradients of the *local cost function* computed at x_i, i.e., to the quantity

$$y_i = \nabla f_i(x_i) = C_i x_i + Q_i \theta_0, \tag{4}$$

where $Q_i := -C_i\Gamma_i$. In these terms, the distributed optimization problem associated to (1) can be cast as follows.

Problem 2.1: Find an update law for x_i, depending only on the local available information given by the quantities (x_j, y_j) for all $j \in \mathcal{N}_i$, such that

$$\lim_{t\to\infty} x_i(t) = \theta^\star = \Sigma\theta_0,$$

for each $i \in \{1, \ldots, N\}$. $\triangle$

Problem 2.1 could be clearly solved in a distributed way through a consensus algorithm by exploiting equation (3). In this paper we focus on distributed optimization algorithms to solve Problem 2.1. It is worth mentioning that in some applications agents may not know C_i, Q_i and θ_0 but just the local *measurement* y_i. Notice that, in view of (4), each matrix C_i is directly linked to the Lipschitz constant of the corresponding local gradient ∇f_i, while the affine terms $Q_i\theta_0$ represent a partial information on θ_0 that, even if accessible via y_i, is not assumed to be known a priori. On this regard, θ_0 is a parameter condensing an information which is not known to the agents.

B. The Gradient Tracking Algorithm

In this subsection, we recall the *gradient tracking algorithm* in its most basic form. For convenience, we first recall the (centralized) gradient method applied to a generic instance of (1). In the (steepest descent) gradient method, a solution estimate $\theta(t)$ is iteratively updated according to[1]

$$\theta^+ = \theta - \gamma\sum_{i=1}^{N}\nabla f_i(\theta),$$

where γ is a constant, positive parameter that is usually called *stepsize*. Convergence results for the class of gradient methods can be found, e.g., in [19].

The gradient tracking distributed algorithm mimics the centralized update by exploiting a twofold consensus-based mechanism to: *(i)* enforce an agreement among the agents' estimates x_i and *(ii)* dynamically track the gradient of the whole cost function through an auxiliary variable $s_i \in \mathbb{R}^d$, called *tracker*. Formally, it reads

$$x_i^+ = \sum_{j\in\mathcal{N}_i} a_{ij}x_j - \gamma s_i \tag{5a}$$

$$s_i^+ = \sum_{j\in\mathcal{N}_i} \tilde{a}_{ij}s_j + \nabla f_i(x_i^+) - \nabla f_i(x_i), \tag{5b}$$

where a_{ij} and $\tilde{a}_{ij}$ are entries of a row stochastic matrix $A \in \mathbb{R}^{N\times N}$ and of a column stochastic matrix $\tilde{A} \in \mathbb{R}^{N\times N}$, respectively, while $\gamma > 0$ is a (constant) stepsize.

Several versions of the gradient tracking algorithm have been analyzed for generic, nonlinear, and possibly constrained versions of problem (1), see, e.g., [1]–[8]. For example, in [7] it is shown that, under strong convexity of the local cost functions f_i, and Lipschitz continuity of their gradients, the sequence $\{(x_1(t), \ldots, x_N(t))\}_{t\geq 0}$ generated by algorithm (5), with $x_i(0)$ arbitrary, $s_i(0) = \nabla f_i(x_i(0))$ and for a sufficiently small stepsize γ, converges to the optimal solution $\theta^\star$ of (1).

[1]As discussed in the Notation paragraph, we omit the time dependence when not strictly necessary.

Remark 2.2: An interesting property of the states s_i is that, by summing over i the update (5b), we can exploit the column stochasticity of the weights $\tilde{a}_{ij}$ to obtain

$$\sum_{i=1}^{N} s_i^{+} - \sum_{i=1}^{N} \nabla f_i(x_i^{+}) = \sum_{i=1}^{N} s_i - \sum_{i=1}^{N} \nabla f_i(x_i). \tag{6}$$

Specifically, condition (6) holds at $t = 0$. Moreover, it also holds for any *consensual* asymptotic value of (x_i, s_i). By assuming $\lim_{t\to\infty} x_i(t) = x_i^{\infty} = x^{\infty}$, for all $i = 1, \dots, N$, it can be shown that the asymptotic value of the tracker is $\lim_{t\to\infty} s_i(t) = s_i^{\infty} = 0$, $\forall\, i$ (recall that weights a_{ij} in (5a) sum up to one, i.e., A is row stochastic). Thus, we have

$$\begin{aligned}\sum_{i=1}^{N} s_i^{\infty} - \sum_{i=1}^{N} \nabla f_i(x_i^{\infty}) &= -\sum_{i=1}^{N} \nabla f_i(x^{\infty}) \\ &= \sum_{i=1}^{N} s_i(0) - \sum_{i=1}^{N} \nabla f_i(x_i(0)).\end{aligned}$$

This, in turn, shows that if the initialization of each s_i is arbitrary, so that the last line is not zero, there is no chance that a consensual asymptotic value is stationary (hence optimal) for problem (1). $\triangle$

The distributed algorithm described by (5) does not enjoy the usual "state-space" structure of dynamical systems, since the updated s_i^{+} depends on x_i^{+}. Thus, we consider the change of variable $z_i := s_i - \nabla f_i(x_i)$, so that algorithm (5) can be equivalently rewritten as

$$x_i^{+} = \sum_{j\in\mathcal{N}_i} a_{ij} x_j - \gamma \left(z_i + \nabla f_i(x_i)\right) \tag{7a}$$

$$z_i^{+} = \sum_{j\in\mathcal{N}_i} \tilde{a}_{ij} z_j + \sum_{j\in\mathcal{N}_i} \tilde{a}_{ij} \nabla f_j(x_j) - \nabla f_i(x_i). \tag{7b}$$

In these new coordinates, the correct initialization becomes $z_i(0)0$, for all $i = 1, \dots, N$. Also, this reformulation does not alter the distributed nature of the algorithm.

Let $x := (x_1, \dots, x_N)$, $z := (z_1, \dots, z_N)$ and $y := (y_1, \dots, y_N)$ and compactly rewrite (7) as

$$\begin{aligned} x^{+} &= \mathbf{A}x - \gamma z - \gamma y \\ z^{+} &= \tilde{\mathbf{A}} z + (\tilde{\mathbf{A}} - I_{dN}) y \\ y &= \nabla F(x), \end{aligned} \tag{8}$$

in which $\mathbf{A} = A \otimes I_d$, $\tilde{\mathbf{A}} = \tilde{A} \otimes I_d$, $z(0) = 0_{dN}$, and $\nabla F(x)$ denotes a column vector stacking the local gradients, i.e., $\nabla F(x) = (\nabla f_1(x_1), \dots, \nabla f_N(x_N))$. Notice that, for the considered quadratic scenario the output map y is (as expected) affine in x and this structure will be exploited later.

III. A System Theoretical Approach to Quadratic Distributed Optimization

In this section, we approach the design of a distributed algorithm solving optimization problem (1) from a system theoretical perspective. Specifically, we approach Problem 2.1 as a generic *set-point control problem*, by giving necessary and sufficient conditions for its solvability. For the sake of presentation, in this part we intentionally avoid dealing explicitly with network constraints. We will discuss constructive choices of the different degrees of freedom that are consistent with the network constraints in Section IV.

A. *The Underlying Control Problem*

The local estimates x_i can be seen as the *state* of N controlled plants

$$x_i^+ = u_i, \qquad i = 1, \dots, N. \tag{9}$$

The control goal consists of finding a suitable control input $u = (u_1, \dots, u_N)$ such that each controlled plant asymptotically converges to the optimal solution $\theta^\star$ of problem (1). We further point out that in this regulation setting the target equilibrium $\theta^\star$ is not available for feedback.

By letting $x := (x_1, \dots, x_N)$ and $y := (y_1, \dots, y_N)$, the overall controlled plant, obtained by stacking the local dynamics (9) and the local measurements (4), reads as

$$\begin{aligned} x^+ &= u \\ y &= Cx + Q\theta_0, \end{aligned} \tag{10}$$

where $C := \operatorname{diag}(C_1, \dots, C_N)$ and $Q := \operatorname{col}(Q_1, \dots, Q_N)$, with C_i and Q_i introduced in (4). Therefore, Problem 2.1 can be recast as follows.

Problem 3.1: Find a (dynamic) controller of the form

$$\begin{aligned} z^+ &= \Phi z + B_x x + B_y y \\ u &= K_z z + K_x x + K_y y, \end{aligned} \tag{11}$$

with state $z \in \mathbb{R}^{Nn_z}$, $n_z \in \mathbb{N}$, and a non-empty set of initial conditions $\mathcal{O} \subset \mathbb{R}^{Nd} \times \mathbb{R}^{Nn_z}$ such that, for each $\theta_0 \in \mathbb{R}^p$, all the trajectories of the "closed-loop system"

$$\begin{aligned} x^+ &= K_x x + K_z z + K_y y \\ z^+ &= \Phi z + B_x x + B_y y, \end{aligned} \tag{12}$$

with $(x(0), z(0)) \in \mathcal{O}$ are bounded and satisfy

$$\lim_{t\to\infty} x_i(t) = \theta^\star = \Sigma\theta_0,$$

for each $i \in \{1, \dots, N\}$. $\triangle$

Restricting the regulator (11) to be linear is motivated by the fact that, except for the affine term $Q\theta_0$ appearing in the output y, the controlled system (10) is *linear*. Thus, Problem 3.1 results in a *linear* set-point control problem that can be solved by a linear regulator. In the same way, linearity implies that we can assume, without loss of generality, that the set of initial conditions $\mathcal{O}$ is an *affine* subspace of $\mathbb{R}^{N(d+n_z)}$ whose bias is parametrized by θ_0, i.e.,

$$\mathcal{O} := P\theta_0 + \mathcal{V}, \tag{13}$$

for some linear subspace $\mathcal{V}$ of $\mathbb{R}^{N(d+n_z)}$ of dimension $n_v \in \mathbb{N}$, and for some matrix $P \in \mathbb{R}^{N(d+n_z)\times p}$ satisfying $\operatorname{Im} P \subset \mathcal{V}^\perp$.

The closed-loop system (12) can be compactly written as

$$\begin{bmatrix} x \\ z \end{bmatrix}^+ = F \begin{bmatrix} x \\ z \end{bmatrix} + G\theta_0, \tag{14}$$

with

$$F := \begin{bmatrix} K_x + K_y C & K_z \\ B_x + B_y C & \Phi \end{bmatrix}, \quad G := \begin{bmatrix} K_y \\ B_y \end{bmatrix} Q.$$

The gradient tracking algorithm (8) exhibits the same closed-loop structure as (14), in which the gradients act as an output feedback action. In the following, we provide necessary and sufficient conditions for the existence of a controller of the form (11) and a set $\mathcal{O}$ of the form (13) solving Problem 3.1.

B. Necessary and Sufficient Conditions

Let $\mathbf{n} := N(d + n_z)$ and $n_v \leq \mathbf{n}$. Consider an n_v-dimensional vector subspace $\mathcal{V}$ of $\mathbb{R}^{\mathbf{n}}$ and let $T \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ be an orthonormal matrix of the form $T = \begin{bmatrix} T_1 & T_2 \end{bmatrix}$, with $T_1 \in \mathbb{R}^{\mathbf{n}\times n_v}$ and $T_2 \in \mathbb{R}^{\mathbf{n}\times(\mathbf{n}-n_v)}$ satisfying

$$\operatorname{Im} T_1 = \mathcal{V}, \quad \operatorname{Im} T_2 = \mathcal{V}^{\perp}. \tag{15}$$

Then, it is easy to see that $\mathcal{V}$ is F-invariant if and only if

$$F' := T^{\top} F T = \begin{bmatrix} F'_I & F'_J \\ 0 & F'_E \end{bmatrix},$$

for some $F'_I \in \mathbb{R}^{n_v\times n_v}$, $F'_J \in \mathbb{R}^{n_v\times(\mathbf{n}-n_v)}$ and $F'_E \in \mathbb{R}^{(\mathbf{n}-n_v)\times(\mathbf{n}-n_v)}$. The matrices F'_I and F'_E represent the restriction of F to $\mathcal{V}$ and $\mathcal{V}^{\perp}$, respectively. These matrices yield to the following definition.

Definition 3.2: The subspace $\mathcal{V}$ is said to be:

- *internally stable* if F'_I is Schur;
- *externally anti-stable* if F'_E has no eigenvalue inside the open unitary disc. △

The forthcoming proposition is the main result of the section and it states necessary and sufficient conditions for the existence of a controller of the form (11) and an initialization set of the form (13) solving Problem 3.1. For simplicity, although not necessary, we restrict the focus to initialization sets with the following additional property.

Definition 3.3: A set $\mathcal{O}$ of the form (13) is said to be an admissible initialization set if $\mathcal{V}$ is F-invariant and externally anti-stable. △

Proposition 3.4: Consider a controller of the form (11) resulting in the closed-loop system (14). Let $\mathcal{O}$ be an admissible initialization set of the form (13), for some n_v-dimensional F-invariant subspace $\mathcal{V}$ of $\mathbb{R}^{\mathbf{n}}$ and some $P \in \mathbb{R}^{\mathbf{n}\times p}$ satisfying $\operatorname{Im} P \subset \mathcal{V}^{\perp}$. Moreover, let $T_2 \in \mathbb{R}^{\mathbf{n}\times(\mathbf{n}-n_v)}$ be an orthonormal matrix satisfying $\operatorname{Im} T_2 = \mathcal{V}^{\perp}$. Then, Problem 3.1 is solved from $\mathcal{O}$ by a controller of the form (11) if and only if

(i) the set $\mathcal{V}$ is internally stable;

(ii) there exists $\Pi \in \mathbb{R}^{\mathbf{n}\times p}$ satisfying

$$\Pi = F\Pi + G \tag{16a}$$

$$\mathbf{I}\Sigma = \begin{bmatrix} I_{Nd} & 0_{Nd\times Nn_z} \end{bmatrix} \Pi \tag{16b}$$

$$0 = T_2^{\top}(\Pi - P). \tag{16c}$$

$\triangle$

Regarding the claim of Proposition 3.4, we observe that equation (16a) expresses the existence, for every $\theta_0 \in \mathbb{R}^p$, of an equilibrium of the closed-loop system (14) given by

$$(x_{\text{eq}}, z_{\text{eq}}) := \Pi\theta_0. \tag{17}$$

Equation (16b), instead, forces such equilibrium to be an optimal solution of problem (1), namely $x_{\text{eq}} = \mathbf{I}\Sigma\theta_0 = \mathbf{I}\theta^\star$. Finally, equation (16c) and the internal stability of $\mathcal{V}$ express the fact that, if the closed-loop system (14) is initialized with $(x(0), z(0)) \in \mathcal{O}$, then the equilibrium point (17) attracts all the closed-loop trajectories.

IV. Gradient Tracking Revisited

In this section, we establish a bridge between the gradient tracking distributed algorithm described in Section II-B and the system theoretical framework discussed in Section III. The design of a distributed optimization algorithm solving problem (1) can be equivalently recast as the problem of finding a regulator of the form (11) which satisfies Problem 3.1 and is *sparse*, in the sense that each control input u_i depends *only on the neighboring information* (x_j, y_j), $j \in \mathcal{N}_i$. Specifically, we show that matrices Φ, B_x, B_y, K_z, K_x, K_y in the controller (11) can be properly chosen to implement a class of gradient tracking algorithms that, among others, includes (8). To this end, we progressively fix the available degrees of freedom in the controller (11) with the aim of satisfying the conditions given in Proposition 3.4.

A. Gradient Tracking as a Control System

First we set the controller dimension equal to the plant dimension, i.e., $n_z = d$. Moreover, we let in the controller (11)

$$\Phi = \tilde{\mathbf{A}}, \qquad B_x = 0, \qquad B_y = \tilde{\mathbf{A}} - I, \qquad K_x := \mathbf{A}, \tag{18}$$

where $\mathbf{A} \in \mathbb{R}^{Nd \times Nd}$ satisfies $\mathbf{A}\mathbf{I} = \mathbf{I}$ and $\tilde{\mathbf{A}} \in \mathbb{R}^{Nd \times Nd}$ $\mathbf{I}^\top \tilde{\mathbf{A}} = \mathbf{I}^\top$, while K_z and K_y are still free. We notice that all the matrices in (18) are sparse, resulting in a controller that can be implemented in a fully distributed way.

The choice (18) results in a closed-loop system (14) with

$$F = \begin{bmatrix} \mathbf{A} + K_y C & K_z \\ (\tilde{\mathbf{A}} - I)C & \tilde{\mathbf{A}} \end{bmatrix}, \quad G := \begin{bmatrix} K_y Q \\ (\tilde{\mathbf{A}} - I)Q \end{bmatrix}. \tag{19}$$

In the following, we investigate conditions on the choice of K_z and K_y such that an admissible initialization set $\mathcal{O}$ and the controller (11) satisfy the assumptions of Proposition 3.4. As a first result we claim the following.

Lemma 4.1: Consider the closed-loop system (14) in the setting described above. Then,

(i) there exists an $(\mathbf{n} - d)$-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^{\mathbf{n}}$ that is F-invariant and externally anti-stable for *all the possible choices* of K_y and K_z;

(ii) there always exist K_y and K_z such that $\mathcal{V}$ is also internally stable.

Proof: We first notice that F in (19) can be decomposed in two terms as

$$F = F_0 + B_0 \begin{bmatrix} K_y C & K_z \end{bmatrix}, \tag{20}$$

where

$$F_0 := \begin{bmatrix} \mathbf{A} & 0 \\ (\tilde{\mathbf{A}} - I)C & \tilde{\mathbf{A}} \end{bmatrix}, \quad B_0 := \begin{bmatrix} I \\ 0 \end{bmatrix}.$$

As a consequence F can be thought of as being obtained by stabilizing the following auxiliary system

$$\mathbf{x}_0^+ = F_0 \mathbf{x}_0 + B_0 \mathbf{u}_0$$

by means of the state-feedback control law $\mathbf{u}_0 := \begin{bmatrix} K_y C & K_z \end{bmatrix} \mathbf{x}_0$. Being F_0 triangular, it holds that $\sigma(F_0) = \sigma(\mathbf{A}) \cup \sigma(\tilde{\mathbf{A}})$. Hence, F_0 has an eigenvalue equal to 1 with algebraic multiplicity $2d$, while all the other eigenvalues lie inside the open unitary disc. It can be shown that a basis for the left-eigenspace of F_0 associated to the eigenvalue 1 is given by the span of $v_1^\top$ and $v_2^\top$ defined as

$$v_1^\top := \begin{bmatrix} v_{11}^\top & 0 \end{bmatrix}, \quad v_2^\top := \begin{bmatrix} 0 & \mathbf{I}^\top \end{bmatrix}, \tag{21}$$

where v_{11} satisfies $v_{11}^\top \mathbf{A} = v_{11}^\top$. We further observe that the left-kernel of $\begin{bmatrix} F_0 - I & B_0 \end{bmatrix}$ is spanned only by $v_2^\top$. Therefore, the stabilizability PBH test ensures that the non-reachable subspace of (F_0, B_0) is a d-dimensional subspace of the eigenspace associated to the eigenvalue 1 and, on the other hand, that the reachable subspace has dimension $n_v = \mathbf{n} - d$. Therefore, point *(i)* follows by taking $\mathcal{V}$ equal to the reachable subspace, and by noticing that its F-invariance and external anti-stability properties cannot be changed via feedback, i.e., by any choice for K_z and K_y.

To show point *(ii)*, we resort to the reachability Kalman decomposition. Consider a transformation matrix $T := \begin{bmatrix} T_1 & T_2 \end{bmatrix}$ with

$$T_1 := \begin{bmatrix} I & 0 \\ 0 & R \end{bmatrix}, \quad T_2 := \frac{v_2}{\sqrt{N}} = \frac{1}{\sqrt{N}} \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix}, \tag{22}$$

where $R \in \mathbb{R}^{Nd \times (N-1)d}$ is such that $RR^\top = I$ and $R^\top \mathbf{I} = 0$. Then, it holds $T^{-1} = T^\top$ and T transforms (F_0, B_0) into $F_0' := T^\top F_0 T$ and $B_0' := T^\top B_0$ of the form

$$F_0' = \left[\begin{array}{cc|c} \mathbf{A} & 0 & F_{J1} \\ -R^\top (I - \tilde{\mathbf{A}})C & R^\top \tilde{\mathbf{A}} R & F_{J2} \\ \hline 0 & 0 & I \end{array}\right], \quad B_0' = \left[\begin{array}{c} I \\ 0 \\ \hline 0 \end{array}\right],$$

for some F_{J1} and F_{J2}. Furthermore by construction the pair

$$(F_I', B_I') := \left(\begin{bmatrix} \mathbf{A} & 0 \\ -R^\top (I - \tilde{\mathbf{A}})C & R^\top \tilde{\mathbf{A}} R \end{bmatrix}, \begin{bmatrix} I \\ 0 \end{bmatrix} \right) \tag{23}$$

is completely reachable, and being C nonsingular, then there always exist gain matrices K_y and K_z satisfying $\begin{bmatrix} K_y C & K_z \end{bmatrix} T = \begin{bmatrix} K_{yI}' & K_{zI}' & K_J' \end{bmatrix}$ such that the matrix $F_I' + B_I' \begin{bmatrix} K_{yI}' & K_{zI}' \end{bmatrix}$ is Schur. Thus, point *(ii)* follows since the latter condition implies that $\mathcal{V}$ is internally stable. ∎

In the rest of the section, we denote by $\mathcal{V}$ the subspace produced by Lemma 4.1. The following result gives a sufficient condition on the choice of K_y and K_z such that equations (16) in Proposition 3.4 admit a solution.

Lemma 4.2: Consider the closed-loop system (14) in the setting described above. Pick K_z and K_y such that

$$K_z = K_y. \tag{24}$$

Then, there exist $\Pi \in \mathbb{R}^{\mathbf{n}\times p}$ and $P \in \mathbb{R}^{\mathbf{n}\times p}$, satisfying $\operatorname{Im} P \subset \mathcal{V}^{\perp}$, such that equations (16) hold.

Proof: Let $\Pi = \operatorname{col}(\Pi_x, \Pi_z)$, with $\Pi_x \in \mathbb{R}^{Nd\times p}$ and $\Pi_z \in \mathbb{R}^{Nd\times p}$. Then, in view of (19), Π solves equations (16a) and (16b) if and only if

$$\begin{aligned}\Pi_x &= (\mathbf{A} + K_y C)\Pi_x + K_z\Pi_z + K_y Q\\ \Pi_z &= (\tilde{\mathbf{A}} - I)C\Pi_x + \tilde{\mathbf{A}}\Pi_z + (\tilde{\mathbf{A}} - I)Q\\ \mathbf{I}\Sigma &= \Pi_x.\end{aligned} \tag{25}$$

By (24) and since $(\mathbf{A} - I)\mathbf{I} = 0$, we can rewrite (25) as

$$\begin{aligned}0 &= K_y(\Pi_z + C\mathbf{I}\Sigma + Q)\\ 0 &= (\tilde{\mathbf{A}} - I)(\Pi_z + C\mathbf{I}\Sigma + Q)\\ \Pi_x &= \mathbf{I}\Sigma.\end{aligned} \tag{26}$$

Therefore, $\Pi_x = \mathbf{I}\Sigma$ and $\Pi_z = -C\mathbf{I}\Sigma - Q$ solve (16a)-(16b).

Finally, as for the existence of P satisfying (16c), we observe that $T_2^\top$ is full rank and $\operatorname{Im} T_2 = \mathcal{V}^{\perp}$. Hence, given Π, equation (16c) is satisfied, e.g., with $P = T_2(T_2^\top T_2)^{-1}T_2^\top\Pi$ that fulfills $\operatorname{Im} P \subset \operatorname{Im} T_2 = \mathcal{V}^{\perp}$. ■

Remark 4.3: While Lemma 4.1 is linked only to a *stability* requirement on the closed-loop system, the choice (24) of Lemma 4.2 represents a constraint ensuring the *existence of an equilibrium which is an optimal solution for the optimization problem* (1). In fact, in order to obtain internal stability of $\mathcal{V}$ we could, for example, choose $K_z = 0$ and K_y any matrix so that $\mathbf{A} + K_y C$ is Schur (which always exists). However, such a choice does not satisfy (24) and, hence, the resulting algorithm would not ensure the existence of an optimal equilibrium for the closed-loop system. △

In the following proposition we merge the previous results to give sufficient conditions on the choice of K_z and K_y so that the trajectories of (14) initialized in $\mathcal{O} = P\theta_0 + \mathcal{V}$ converge to a solution of Problem 3.1.

Proposition 4.4: Consider the closed-loop system (14) in the setting described above. Let $K_z = K_y$ be such that F in (19) has all the eigenvalues but d inside the open unitary disc. Then, Problem 3.1 is solved from $\mathcal{O}$, in the sense that all the trajectories of the closed-loop system (14) originating in $\mathcal{O}$ are bounded and $\lim_{t\to\infty} x_i(t) = \theta^\star$, $\forall\, i = 1, \ldots, N$. △

Proof: In view of Lemma 4.1, there exists an F-invariant and externally anti-stable subspace $\mathcal{V}$. Moreover, it is also internally stable whenever K_z and K_y are such that F has all the eigenvalues but d inside the open unitary disc. In view of Lemma 4.2, if $K_z = K_y$, then there exist Π and P such that steady-state condition (16) hold. Hence, the claim follows by Proposition 3.4. ■

Finally, we notice that the choice of K_z and K_y in Proposition 4.4 might not satisfy the network constraints. In the following, we discuss how the usual practice in distributed optimization of selecting a common stepsize $\gamma > 0$

for all the agents, is consistent with Proposition 4.4, provided that γ is taken sufficiently small. In our framework, this is achieved by setting $K_y = K_z = -\gamma I$. In this way we complete the result of the section by showing that K_z and K_y, fulfilling both the assumptions of Proposition 4.4 and the network constraints, always exist. The feasibility of this choices follows as a particular case of the following result.

Proposition 4.5: Consider the closed-loop system (14) in the setting described above and let $K_z = K_y = -\Lambda$, with $\Lambda \in \mathbb{R}^{Nd \times Nd}$ diagonal and positive definite. Then, there exists $\gamma^\star > 0$ such that, if all the eigenvalues of Λ lie in $(0, \gamma^\star)$, Problem 3.1 is solved from $\mathcal{O}$, in the sense that all the trajectories of the closed-loop system (14) originating in $\mathcal{O}$ are bounded and $\lim_{t\to\infty} x_i(t) = \theta^\star$, $\forall\, i = 1, \ldots, N$. △

B. Remarks About the Proposed Approach

We start by pointing out some aspects related to the initialization. We observe that, if K_y is nonsingular, then the choice of Π_z satisfying (26) is unique and it is given by $\Pi_z = -(C\mathbf{I}\Sigma + Q)$. Thus, recalling the definition of Σ in (3), it holds $\mathbf{I}^\top \Pi_z = -\mathbf{I}^\top (C\mathbf{I}\Sigma + Q) = 0$. Moreover, we have shown that we can set the matrix $T_2^\top = v_2^\top = \begin{bmatrix} 0 & \mathbf{I}^\top \end{bmatrix}$. Thus, equation (16c) leads to $\begin{bmatrix} 0 & \mathbf{I}^\top \end{bmatrix} P = \mathbf{I}^\top \Pi_z = 0$. This means that the admissible initialization set $\mathcal{O}$ coincides with $\mathcal{V}$, i.e., the distributed algorithm works *only if* initialized in the reachable subspace $\mathcal{V}$, which means that $(x(0), z(0))$ has to be chosen so that $\sum_{i=1}^N z_i(0) = 0$. This, in turn, is consistent with Remark 2.2. However, we point out that $\sum_{i=1}^N z_i(0) = 0$ is necessary only if $\ker K_y \cap \ker(\tilde{\mathbf{A}} - I) = \{0\}$, as otherwise different choices of Π_z might be possible.

We underline that the only parameters of the problem (1)-(2) that need to be known for the design of the gains K_y and K_z (fulfilling the stability requirement of Proposition 4.4) are the matrices C_i. Nevertheless, due to the continuity of the eigenvalues of the closed-loop matrix F with respect to variations in K_y and K_z, we also observe that whenever internal stability of $\mathcal{V}$ is ensured for a "nominal" value C_i° of C_i, it also holds for all the actual values of C_i in a sufficiently small open neighborhood of C_i°.

More in general, well-known results in the context of (hybrid) dynamical systems (see, e.g., [20, Proposition 6.34]), show that any algorithm of the form (11) fulfilling the conditions of Proposition 3.1 is "robust" with respect to parameter perturbations and measurement noise. That is, for sufficiently small perturbations and noise, boundedness of the closed-loop trajectories is preserved, and the asymptotic error from the optimal solution is related to the noise bound.

Finally, we underline how well-known arguments on homogeneous approximations of nonlinear systems (see, e.g., [20, Theorem 9.11]) can be used to show that the global (in $\mathcal{O}$) result of Proposition 3.1 implies a *local* (in $\mathcal{O}$) result when sufficiently regular nonlinearities comes into play. This, in turn, permits to extend the presented results "locally" to optimization problems of the form (1)-(2) with nonlinear, strongly convex functions and smooth f_i.

V. Conclusions

In this paper we proposed a system theoretical approach to analyze a class of gradient tracking algorithms for distributed quadratic optimization. We formulated the design of a distributed algorithm as the design of a (linear) dynamic regulator solving a set-point control problem. We highlighted structural properties of the designed regulator and we showed that they are fulfilled by the gradient tracking. Moreover, we proved how lack of reachability of the closed-loop system imposes conditions on the initialization of distributed algorithms with this structure. The proposed system theoretical perspective suggests that robustness arguments, customary in control theory, can be used to draw similar conclusions on the optimization algorithms. Finally, these results pave the way to more general technical tools for the analysis of nonlinear, distributed optimization problems.

REFERENCES

[1] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Trans. on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[2] D. Varagnolo, F. Zanella, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-Raphson consensus for distributed convex optimization," *IEEE Trans. on Autom. Control*, vol. 61, no. 4, pp. 994–1009, 2016.

[3] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[4] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, 2018.

[5] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Trans. on Autom. Control*, vol. 63, no. 2, pp. 434–448, 2018.

[6] C. Xi, R. Xin, and U. A. Khan, "ADD-OPT: Accelerated distributed directed optimization," *IEEE Trans. on Autom. Control*, vol. 63, no. 5, pp. 1329–1339, 2018.

[7] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 315–320, 2018.

[8] G. Scutari and Y. Sun, "Distributed nonconvex constrained optimization over time-varying digraphs," *Mathematical Programming*, vol. 176, no. 1-2, pp. 497–544, 2019.

[9] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. on Autom. Control*, vol. 54, no. 1, pp. 48–61, 2009.

[10] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. on Autom. Control*, vol. 55, no. 4, pp. 922–938, 2010.

[11] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[12] S. S. Kia, B. Van Scoy, J. Cortés, R. A. Freeman, K. M. Lynch, and S. Martínez, "Tutorial on dynamic average consensus: the problem, its applications, and the algorithms," *preprint arXiv:1803.04628*, 2018.

[13] J. Wang and N. Elia, "Control approach to distributed optimization," in *IEEE Allerton Conf. on Communication, Control, and Computing*, 2010, pp. 557–561.

[14] ——, "A control perspective for centralized and distributed convex optimization," in *IEEE Conf. on Decision and Control and European Control Conf. (CDC)-(ECC)*, 2011, pp. 3800–3805.

[15] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 57–95, 2016.

[16] B. Hu and L. Lessard, "Control interpretations for first-order optimization methods," in *IEEE American Control Conf. (ACC)*, 2017, pp. 3114–3119.

[17] A. Sundararajan, B. Hu, and L. Lessard, "Robust convergence analysis of distributed optimization algorithms," in *IEEE Allerton Conf. on Communication, Control, and Computing*, 2017, pp. 1206–1212.

[18] T. Hatanaka, N. Chopra, T. Ishizaki, and N. Li, "Passivity-based distributed optimization with communication delays using PI consensus algorithm," *IEEE Trans. on Autom. Control*, vol. 63, no. 12, pp. 4421–4428, 2018.

[19] D. P. Bertsekas, *Nonlinear programming*. Athena scientific, 1999.

[20] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems. Modeling, Stability, and Robustness*. Princeton University Press, 2012.